Workflows and Decision Tables for Flexible Early Warning Systems

Felix Riedel
Fraunhofer IOSB
felix.riedel@iosb.fraunhofer.de

Fernando Chaves
Fraunhofer IOSB
fernando.chaves-salamanca@iosb.fraunhofer.de

ABSTRACT

Today's decision support systems for crisis management are mostly designed to support a fixed process that integrates a given set of information sources. This means policies that govern the crisis management process are tightly integrated with the implementation, which makes it hard to adapt the system to changing requirements. Modern systems are expected to be adaptable and need to evolve along with the availability of new information sources and changing business processes. Previous work suggested using workflow systems to manage crisis management processes. Current approaches that use workflow systems are not end-user friendly or not flexible enough. In this paper we present our approach that combines workflows and decision tables for creating more flexible decision support systems. While workflows are used to orchestrate services and implement information logistics in the decision support processes, embedded rule sets are used to provide flexibility and adaptability of workflows. The rule sets are authored using decision tables which are an easy-to-use representation that allows end-users to express rules in an intuitive way.

Keywords

Flexibility, usability, workflows, rules, decision tables, decision support

INTRODUCTION

An essential part of early warning systems and systems for crisis management are communication-driven decision support systems that facilitate communication and collaboration for decision making. Often official policies specify how different organizations collaborate and what information is communicated to whom. For early warning systems it is crucial that information is exchanged in a timely manner and all participants get exactly the information they need to fulfil their role in the crisis management process. Information technology obviously lends itself to automate parts of the process. We have experienced however that in current operational systems the information logistics processes are hard-coded even though they are subject to change. In addition, systems are tailored to the policies and requirements of a certain organization and changes can require major refactoring. We seek to develop a system that can be deployed and adapted to multiple organizations with different policies. A major requirement for such a system is that changes can be applied locally without affecting larger parts of the system. In addition to the flexibility regarding changes in policies and processes, we also want a system that can evolve. This means when new information sources become available, it should be possible to integrate and use these in the decision process.

In general this kind of flexibility comes with a significant increase of complexity. This means only IT professionals can maintain a system that can be reconfigured and adapted; end-users are unable to utilise the provided flexibility. In the business world there is a similar problem. Business processes change and evolve over time and IT systems have to be readjusted constantly. A standard solution for this is to use business process management systems (BPMS) or workflow management systems (WfMS) that execute workflows modelled using graphical notations, such as BPMN2 (Business Process Model and Notation 2.0). Graphical notations are easier to grasp and provide a better overview of what happens in a process and therefore it is easier to revise graphically modelled workflows than code written in a general purpose programming language.

Sell and Braun (2009) pointed out the similarity between business processes and emergency plans and proposed the use of a WfMS for automating and managing emergency plans. We also believe that decision support and early warning systems can profit from the business process technology. However, the flexibility and usability of current WfMS are limited. Jansen, Lijns, and Plasmeier (2010) found that contemporary workflow systems are too rigid and are not usable if only a rough sketch of actions is given. In addition to these deficiencies the current workflow systems and their notations are still not suitable for end-users. The general idea is that a graphical notation like BPMN2 is understandable by end-users and the users are therefore able to model and adapt workflows themselves. Although a graphical notation is a great tool to improve communication between end-users and developers, authoring executable workflows is still something an average end-user cannot carry out alone.

In this paper we show that decision tables are an intuitive tool that can be used by domain experts to express sets of rules that can be interpreted automatically at runtime. Furthermore, the combination of workflows and decision tables creates a framework that opens possibilities for flexible and adaptable workflows.

DEFINITIONS AND BACKGROUND

This section gives a short introduction into workflows and decision tables. Both are used in our approach which is described in the next section.

Workflows

Originally, a workflow meant a specified flow of work which was implemented by persons who passed paper from one to another in order to exchange and enrich information on a certain matter. With the software implementation of workflows documents as well as procedures to exchange documents are digitalized and certain activities are automated. In recent years the term business process became popular and different viewpoints exist about how workflows and business processes relate to each other. In this document we follow the terminology of the Workflow Management Coalition (1999) which defines a workflow as "The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules". Such a workflow consists of several steps or activities that need to be executed following a defined and structured sequence. The term task is used to refer to a step inside a workflow.

Decision Tables

	Condition entries				**Action entries**	
	Variable 1	Variable 2	Variable 3		Action 1	Action 2
Rule 1	> 10	-	-		X	
Rule 2	> 0	== 0	-			
Rule 3	> 0	> 0	< 5		X	X
Rule 4	< 0	-	< 0			X

Table 1. A decision table with four rules.

Decision tables have been around for decades but few developers are familiar with them or they have forgotten about this simple but powerful technique for dealing with elaborate rule sets (Pollack, 1963; Thomas, 2005). A decision table is basically a set of decision rules, which consist of a set of conditions and a set of actions that are performed if the conditions are met. In a horizontal layout each row corresponds to a single decision rule (cp. Table 1). A decision table is only a different representation of propositional if-then-rules. The major advantage of decision tables is that they are significantly easier to deal with than other representations of propositional rules. An empirical evaluation of the comprehensibility of decision tables, decision trees and textual representation of rules showed that decision tables perform significantly better than their competitors on all evaluation criteria (Huysmans, 2010). The criteria were accuracy of the interpretation, response time and answer confidence for a set of problem-solving tasks. Also, a majority of the users found decision tables the easiest representation format to work with.

WORKFLOWS AND DECISION TABLES FOR TSUNAMI EARLY WARNING

As mentioned earlier and proposed by others (Sell et al., 2009; Jansen et al., 2010) a workflow engine is used to orchestrate services, to gather information from different sources and to structure the human involvement in decision support processes. The same functionality could also be provided by scripts or programs written in a general purpose programming language, but in general they lack several advantages that workflows can provide.

The graphical notation of workflows allows non-technical personnel to understand what a process does, which actors are involved and what the single tasks of the process are. This facilitates the communication between domain experts and IT experts. Domain experts know the policies and know the workflows an organization wants to implement, but IT experts are required to create an executable workflow that handles the intricacies of service invocation and data flow. With a common notation such as BPMN2 domain experts can create a sketch of a workflow, IT experts can refine the workflow using the same notation and the result is a workflow that both, domain and IT expert, understand.

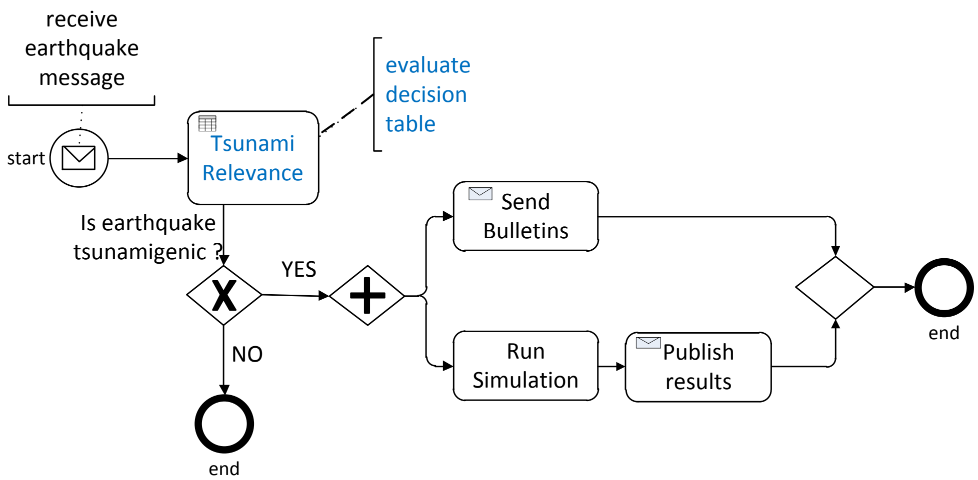

Figure 1. Simplified tsunami warning workflow with integrated decision table (BPMN2 notation).

At run time the notation can be used to provide a graphical representation of the running process which illustrates the status of a workflow; for instance, the currently executed task is highlighted in the workflow diagram. In the same spirit logging and tracing capabilities of a workflow engine can be used to keep a history of executed workflows and tasks. The history can be used for debugging, post mortem analysis, monitoring or optimization of processes. In hand-coded processes this functionality has to be implemented with additional efforts.

Figure 1 shows a simplified version of a workflow that is used in the tsunami early warning system that we are working on in the scope of a current European research project. The workflow is triggered by an earthquake event that is published by the seismic system via a message oriented-middleware (MOM). The earthquake message contains all relevant earthquake parameters such as epicentre location, depth and magnitude. Based on these earthquake parameters the first task determines whether the earthquake is potentially tsunamigenic or not. If the earthquake has potential for a tsunami bulletins are sent and a simulation service is invoked in parallel in order to acquire more information about a possible tsunami.

Depth	Epicentre Location	Mw	Tsunami Potential	Local Bulletin	Regional Bulletin	Basin-wide Bulletin
< 100 km	Offshore or close to the coast (< 40 km inland)	5.5 to 6.0	Weak potential of local tsunami	Advisory	Information	Information
		6.0 to 6.5	Potential of destructive local tsunami < 100 km	Watch	Advisory	Information
	Offshore or close to the coast (< 100 km inland)	6.5 to 7.0	Potential of destructive regional tsunami < 400 km	Watch	Watch	Advisory
		> 7.0	Potential of destructive tsunami in the whole basin < 100 km	Watch	Watch	Watch
> 100 km	Offshore or inland (< 100 km)	> 5.5	Nil	Information	Information	Information

Table 2. The ICG/NEATWS tsunami decision matrix for the Mediterranean.
Source: ICG/NEAMTWS Interim Operational Users Guide.

In the Mediterranean the distances between potentially tsunamigenic sources and the coast are rather small; hence, the relevance assessment that is done in the first step of the workflow should be as fast as possible. Results must be on hand within less than 5 minutes after the earthquake's origin time. This is why the *Intergovernmental Coordination Group* for the *North Eastern Atlantic, the Mediterranean and connected Seas Tsunami Warning System* (ICG/NEAMTWS) provides decision matrices that the staff of the Tsunami Watch Providers can use to quickly determine the tsunami potential and the type of bulletin that need to be send (Intergovernmental Oceanographic Commission, 2011). The current tsunami decision matrix for the Mediterranean is shown in Table 2. As one can see the decision matrix is a decision table with conditions based on the earthquake parameters on the left and the tsunami potential and the bulletin types as output or action on the right. The fact that decision tables in their paper form are already used in the domain of tsunami early warning supports our assumption that decision tables are good fit when it comes to providing end-users with an interface to author, revise and review rule sets.

In the example above a decision table was used to represent a rule set that determines tsunami potential and bulletin types given the earthquake parameters; correspondingly, rules and therefore decision tables can be used to automate all kinds of policies for information logistics. By outsourcing the rules from the workflows we create a number of advantages. First, changes to rules and policies can be made by editing decision tables, a representation they are a familiar with. Second, decision tables can be managed and versioned independent from workflows. This means they can also be tested and reviewed independent from the workflows they are used in.

Implementation

The current prototype system is implemented in Java using a workflow engine and a rule engine. As workflow engine we use Activiti, a lightweight, open-source workflow engine which supports BPMN2. JBoss Drools Expert is used to implement the decision tables. For the workflow engine we implemented custom tasks that fetch rule sets from a rule management system and evaluate the rule set using the rule engine.

Workflows can be edited using a BPMN 2.0 editor and deployed using a web-service or a web-based user interface. A web-based console allows managing, starting and stopping workflow processes as well as monitoring running workflows. Figure 2 shows how the user interface displays a running workflow. The currently executed task is highlighted and additional information, such as execution time and context information, is also shown (not in the screenshot).

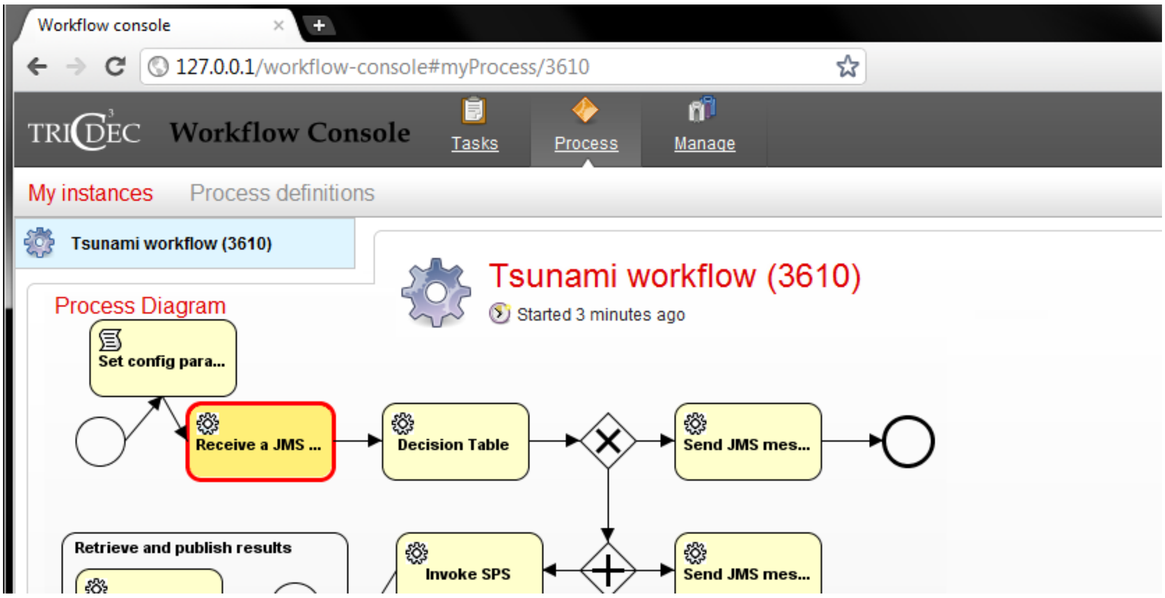

Figure 2. Web-based workflow console showing a running workflow. Currently executed task is highlighted in red.

Decision tables can be edited using a web-based interface or using a standard spreadsheet application, such as Microsoft Excel. When submitted to the rule management system, the decision tables are transformed to a simple rule set which can be interpreted by the rule engine. The advantage of using a rule engine instead of using a simpler implementation of decision tables is that the rule engine uses a clever pattern matching algorithm based on the Rete algorithm (Forgy, 1982) to efficiently evaluate rules. The system especially benefits from this implementation when the rule set is considerable large or if rules use a lot input variables.

CONCLUSION AND OUTLOOK

The presented approach shows how crisis management and early warning systems could benefit from the combination of workflows and decision tables. Decision tables prove to be a good representation to enable end-users to author and modify rule sets that are machine-readable and can be automatically interpreted. In future we plan to use decision tables not only for business rules, but also to implement adaption patterns as proposed by (Döhring, Zimmermann & Karg, 2011). Rule-based adaption patterns allow adapting segments of workflows at runtime in order to be flexible regarding to changes in context or policy.

Our prototype system is usable, but still has some hard-wired parts. These parts deal primarily with providing the rule engine with the necessary information (input facts) to evaluate decision tables. Hence, current work deals with closing the gap between variables that are used within decision tables and information sources. So called semantic decision tables are being developed which allow using concepts of an ontology to describe input variables of decision tables. At design time the concepts will be resolved to information sources which were previously annotated using the same ontology. The result will enable users to use any information that was modelled in the underlying ontology. Discovery of information and data sources will also be build on semantic descriptions that reference a common ontology.

Furthermore, the propagation of trust and belief information is another topic that is researched. This becomes especially important when the decisions basis includes external or less trusted information sources.

ACKNOWLEDGEMENTS

This work is done as part of the TRIDEC project which is supported by the European Commission under the 7th Framework Programme (ICT-2009.4.3 Intelligent Information Management Project Reference: 258723).

REFERENCES

1. Döhring, M., Zimmermann, B., & Karg, L. (2011) Flexible workflows at design- and runtime using BPMN2 adaptation patterns. In W. Abramowicz (Ed.), *14th International Conference on Business Information Systems* (pp. 25–36). Poznan: Springer.
2. Forgy, C. (1982). Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial intelligence*, *19*(1), 17-37. doi:10.1016/0004-3702(82)90020-0
3. Huysmans, J., Dejaeger, K., Mues, C., Vanthienen, J. & Baesens, B. (2010) An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems.* doi:10.1016/j.dss.2010.12.003
4. Intergovernmental Oceanographic Commission (2011) Reducing and Managing the Risk of Tsunamis. *IOC Manuals and Guide*, **57**, UNESCO, Paris.
5. Jansen, J. M., Lijnse, B. & Plasmeier, R. (2010) Towards Dynamic Workflow Support for Crisis Management. In: *Proceedings of the 7th International ISCRAM Conference*
6. Pollack, S. L. (1963). Analysis of the Decision Rules in Decision Tables. *Memorandum RM-3669-PR.* RAND Corp., Santa Monica, California.
7. Sell, C. and Braun, I. (2009) Using a workflow management system to manage emergency plans. In: *Proceedings of the 6th International ISCRAM Conference*
8. Thomas, D. (2005) Agile Programming: Design to Accommodate Change. *IEEE Software*, *22*(3).
9. Workflow Management Coalition (1999) WfMC: Terminology & Glossary, Document Number WFMC-TC-1011, Document Status - Issue 3.0.